



Oxidising RPKI

Martin Hoffmann
– RIPE 77 –



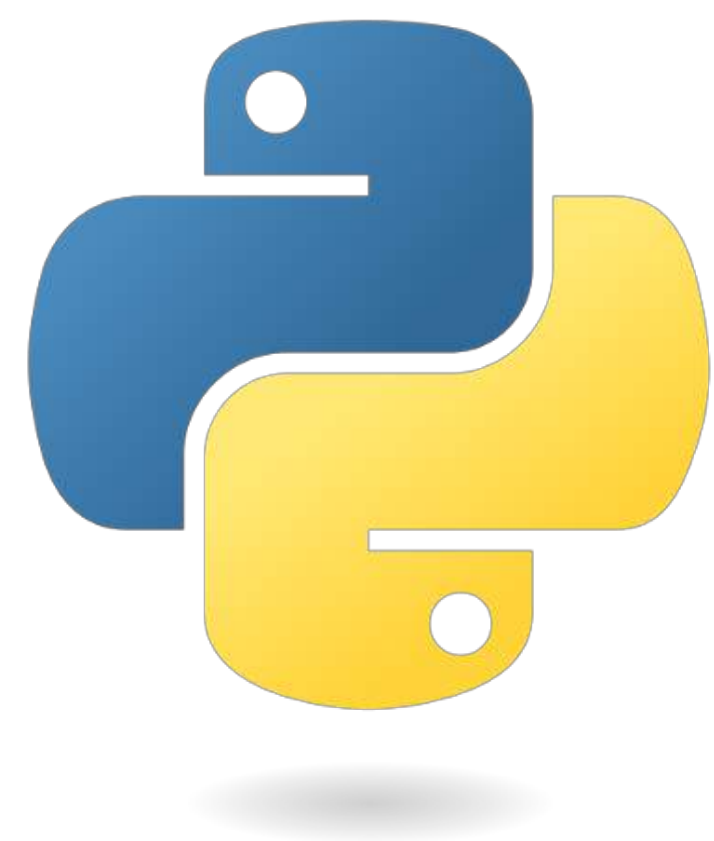
THE
C
PROGRAMMING
LANGUAGE

Brian W. Kernighan • Dennis M. Ritchie

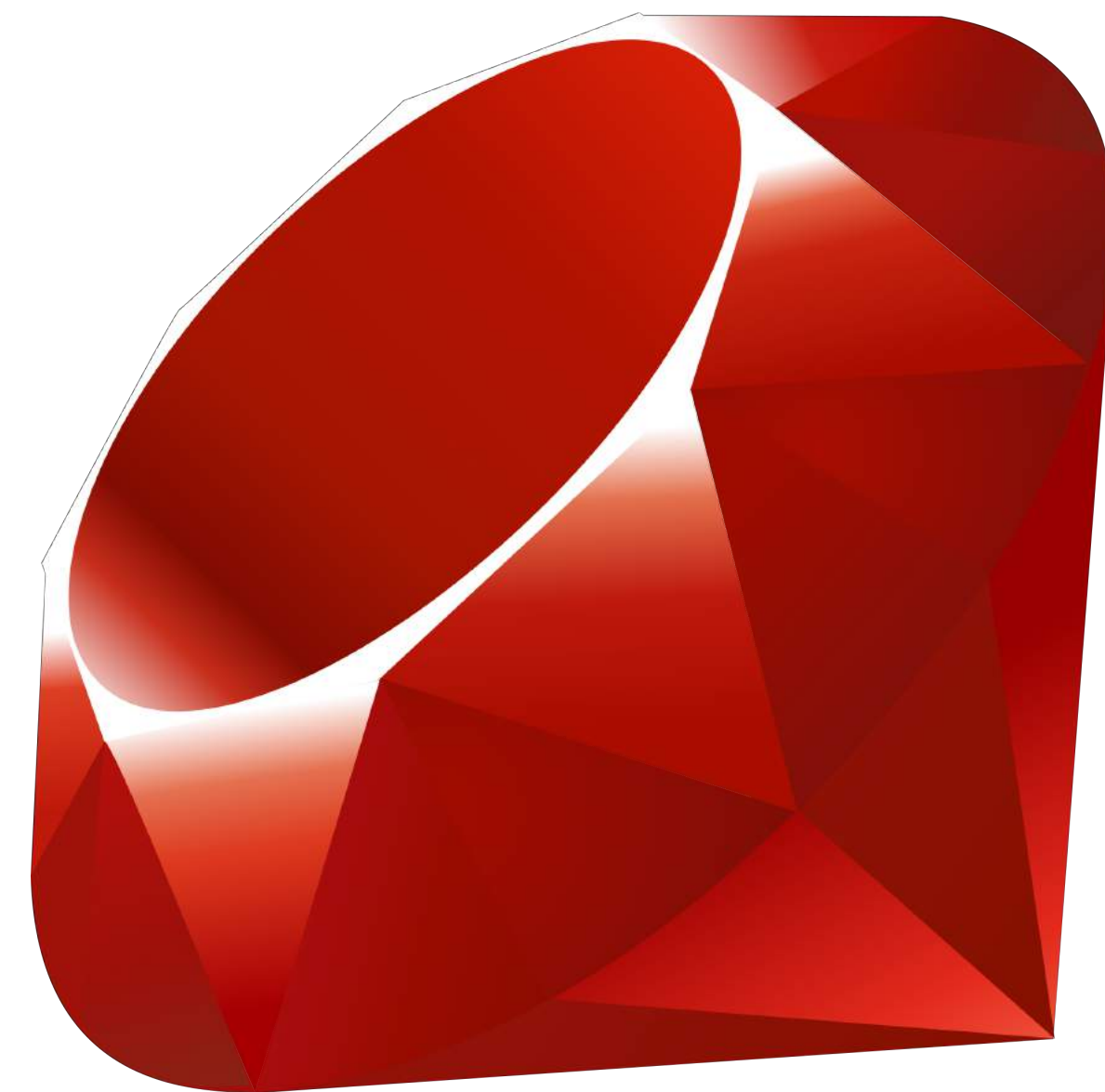
PRENTICE HALL SOFTWARE SERIES



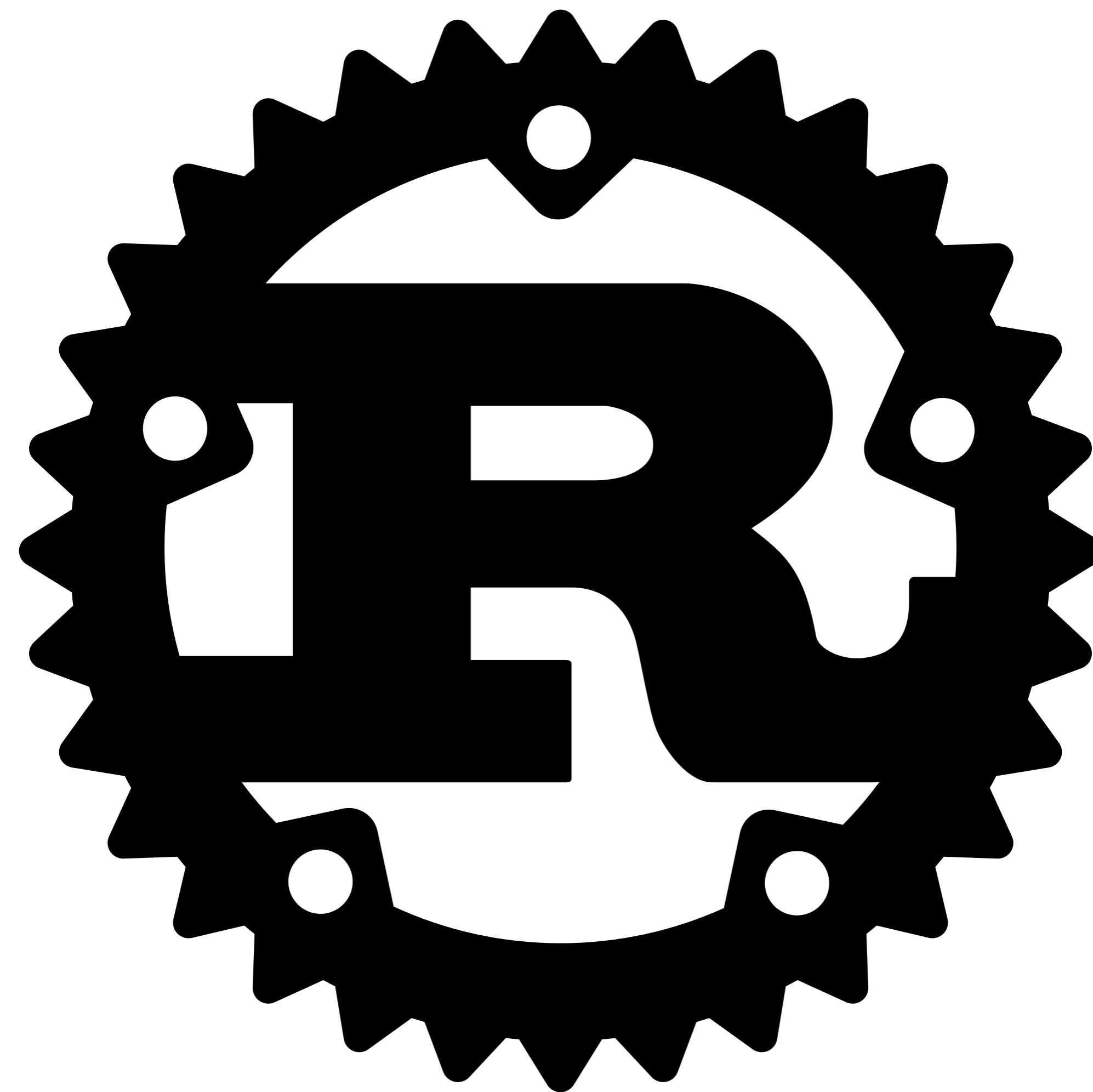
JavaTM

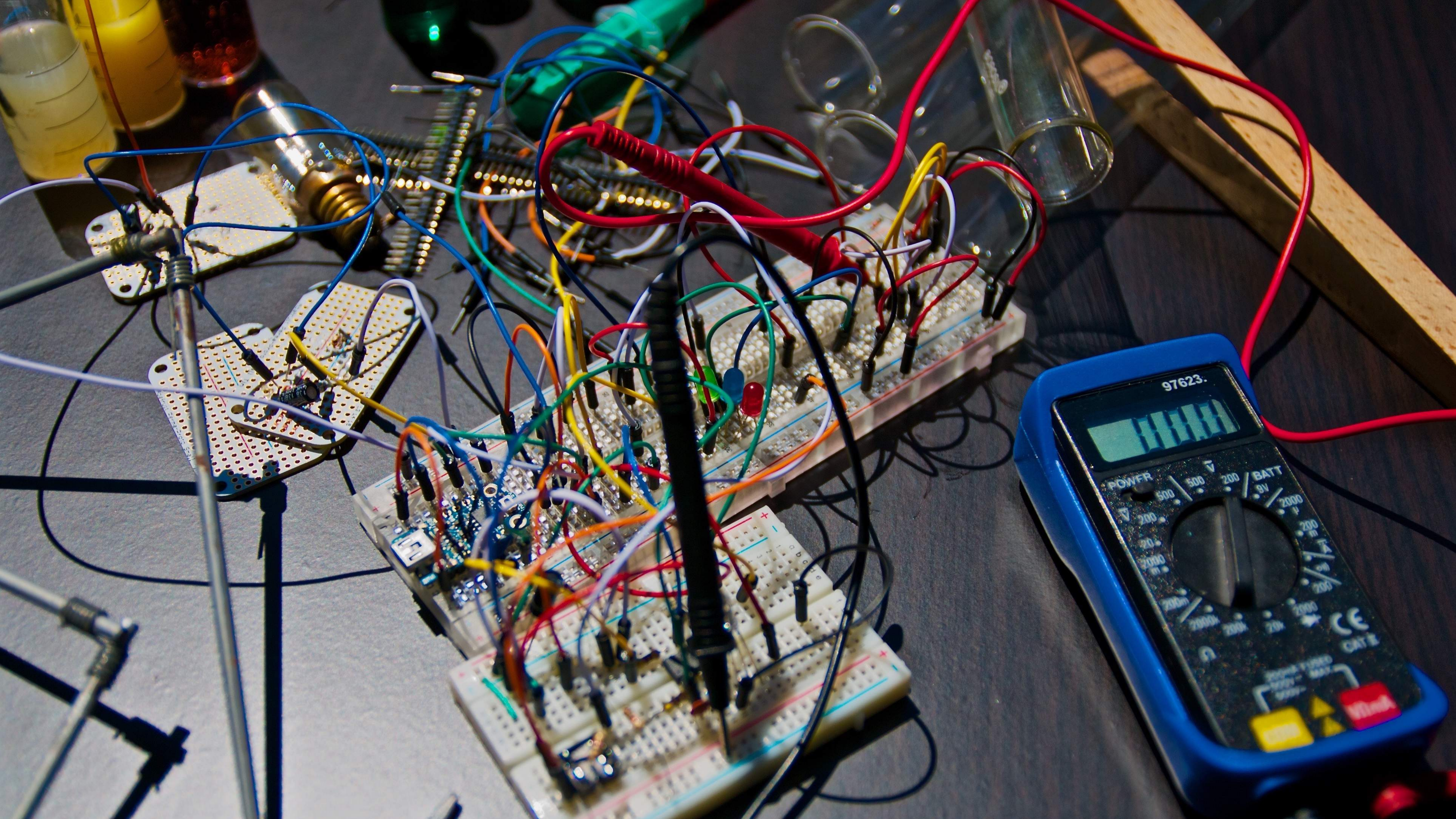


pythonTM



 **GO**





The Why for Engineers

- what everyone always says
- great type system
- the build system
- FFI
- great, friendly community
- the challenge of a lifetime

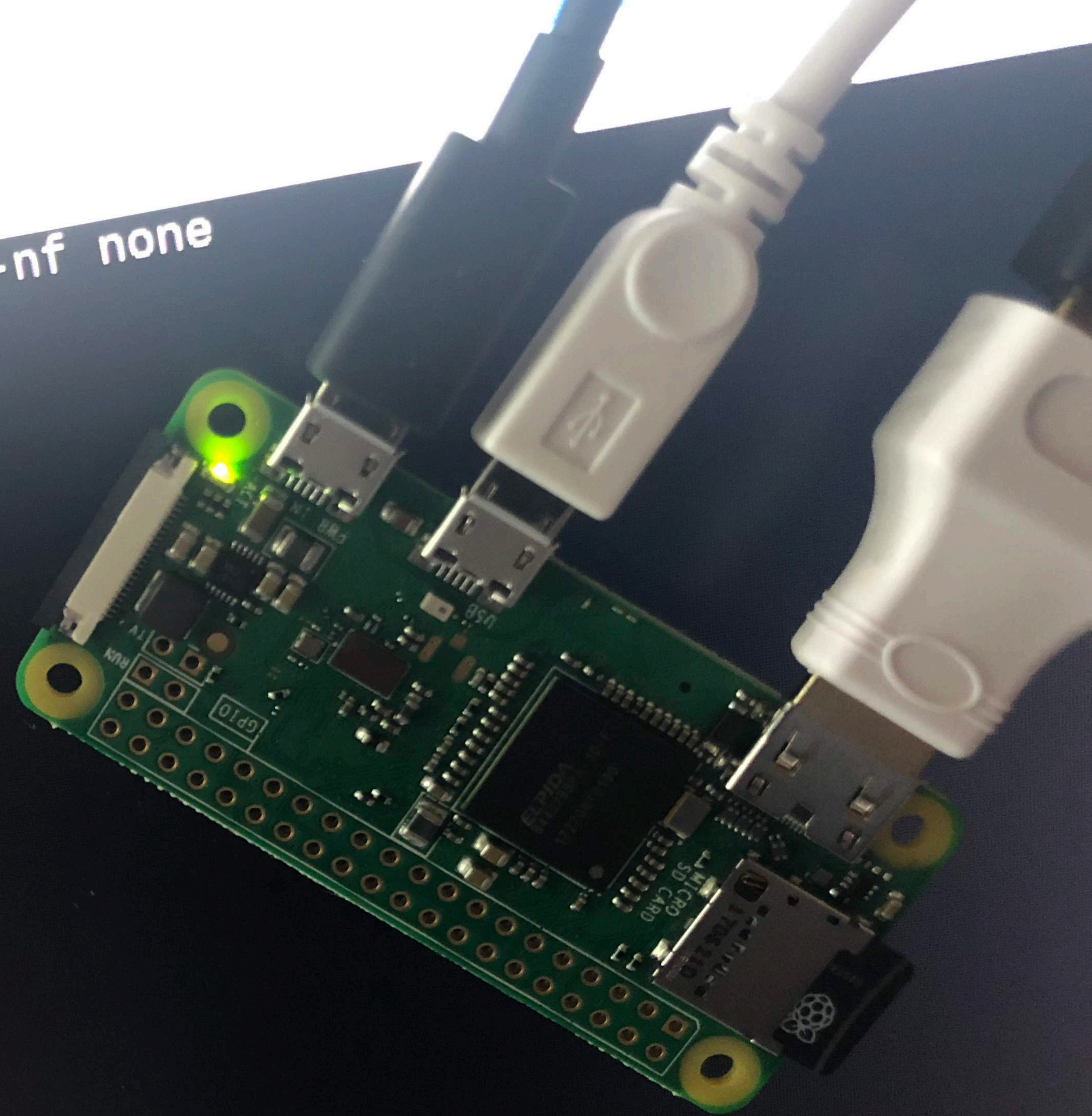
The Why Nots for Managers

- Maturity (Will it break?)
- Continuity (Will it be around in five years?)
- Ecosystem (Do we have to write everything ourselves?)
- Developers, developers, developers

```
pi@raspberrypi:~$ time ./routinator -nf none
```

```
real    4m58.126s  
user    4m43.880s  
sys     0m6.080s
```

```
pi@raspberrypi:~$
```





NLnetLabs/ber-rs

- ~4000 loc
- 2 dependent crates (octet sequences, error handling)

NLnetLabs/rpki-rs

- ~7000 loc
- 11 direct dependencies (crypto, XML, logging, ...)

NLnetLabs/routinator

- ~3000 loc
- 14 direct dependencies (network IO, JSON, parallel processing, time, ...)
- 113 total dependencies
- release build executable 3.4M (2.5M stripped), debug build 16.2M

Interested?

-  <https://nlnetlabs.nl/projects/rpki/about/>
-  <https://github.com/nlnetlabs/routinator>
-  rpki@nlnetlabs.nl
-  [@nlnetlabs](https://twitter.com/nlnetlabs)